AF/2193
$ SFW

OIPE
IAP35
SEP 28 2007
PATENT & TRADEMARK OFFICE

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 10019980-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Brian FAHS et al.	**Confirmation No.:** 7384
Application No.: 10/016,949	**Examiner:** Insun Kang
Filing Date: December 13, 2001	**Group Art Unit:** 2193

Title: METHOD AND SYSTEM TO ANALYZE INLINED FUNCTIONS

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on 7/26/2007.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

□ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

□ 1st Month $120 □ 2nd Month $450 □ 3rd Month $1020 □ 4th Month $1590

□ The extension fee has already been filed in this application.

☑ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of $500. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

☑ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450

Date of Deposit: 09/26/2007

OR

□ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.

Date of facsimile:

Typed Name: Ilene L. Fish

Signature: Ilene L. Fish

Respectfully submitted,

Brian FAHS et al.

By

John P. Wagner, Jr.

Attorney/Agent for Applicant(s)

Reg No. : 35,398

Date : 09/26/2007

Telephone : 408-377-0500

OIPE IAP35
SEP 28 2007
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellant:	Fahs et al.	Patent Application	
Application No.:	10/016,949	Group Art Unit:	2193
Filed:	December 13, 2001	Examiner:	Kang, I.

For: METHOD AND SYSTEM TO ANALYZE INLINED FUNCTIONS

APPEAL BRIEF

10/01/2007 FMETEKI1 00000073 082025 10016949
01 FC:1402 500.00 DA

Table of Contents

I. Real Party in Interest

The assignee of the present application is Hewlett-Packard Development Company, L.P.

II. Related Appeals and Interferences

There are no related appeals or interferences known to the Appellant.

III. Status of Claims

Claims 1-24 are rejected. This Appeal involves Claims 1-24.

IV. Status of Amendments

All proposed amendments have been entered. An amendment subsequent to the Final Action has not been filed.

V. Summary of Claimed Subject Matter

Independent Claim 1 of the present application pertains to a computer-implemented method for examining an inlined function using a performance analysis tool; independent Claim 7 of the present application pertains to a computer-readable medium embodying instructions that cause a computer to perform a method for examining an inlined function using a performance analysis tool; and independent Claim 13 of the present application pertains to an apparatus for examining an inlined function using a performance analysis tool.

In Claim 1, "a computer-implemented method for examining an inlined function using a performance analysis tool" is recited. At least one embodiment is depicted in Figures 2 and 3 and described at least on page 8, line 35 - page 9, line 40. For example, "identifying an inlined function in source code, wherein said source code is for a binary executable," as recited in Claim 1, is described at least in 202 of Figure 2 and on page 8, lines 35-40. "[i]nserting a breakpoint at the start of said inlined function in said binary executable," as recited in Claim 1, is described throughout the Specification including at least page 9, lines 11-17. "[r]eplacing said inlined function with a long branch to a shared memory probe code sequence," as recited in Claim 1, is described throughout the Specification including at least page 9, line 37 through page 10 line 10.

In Claim 7, "A computer-readable medium embodying instructions that cause a computer to perform a method for examining an inlined function using a performance analysis tool" is recited. At least one embodiment is depicted in Figures 2 and 3 and described at least on page 8, line 35 - page 9, line 40. For example, "identifying an inlined function in source code, wherein said source code is for a binary executable," as recited in Claim 7, is described at least in 202 of Figure 2 and on page 8, lines 35-40. "[i]nserting a breakpoint at the start of said inlined function in said binary executable," as recited in Claim 7, is described throughout the Specification including at least page 9, lines 11-17. "[r]eplacing said inlined function with a long branch to a shared memory probe code sequence," as recited in Claim 7, is described throughout the Specification including at least page 9, line 37 through page 10 line 10.

In Claim 13, "An apparatus for examining an inlined function using a performance analysis tool" is recited. At least one embodiment is depicted in Figures 2 and 3 and described at least on page 8, line 35 - page 9, line 40. For example, "means for identifying an inlined function in source code, wherein said source code is for a binary executable," as recited in Claim 13, is described at least in 202 of Figure 2 and on page 8, lines 35-40. "[m]eans for inserting a breakpoint at the start of said inlined function in said binary executable," as recited in Claim 13, is described throughout the Specification including at least page 9, lines 11-17. "[m]eans for replacing said inlined function with a long branch to a shared memory probe code sequence," as recited in Claim 13, is described throughout the Specification including at least page 9, line 37 through page 10 line 10.

VI. Grounds of Rejection to Be Reviewed on Appeal

1. Claims 1-24 are rejected under 35 U.S.C. §103(a) as being unpatentable over Hundt ("HP Caliper-An Architecture for Performance Analysis Tools," 8/2000) in view of Srivestava et al. (U.S. Patent No. 5,963,740), hereinafter "Srivastava." (Appellants respectfully note that the reference is dated August 2000 in the Office Action, while the reference itself carries a date of October 2000.)

VII. Argument

1. Whether Claims 1-24 are unpatentable under 35 U.S.C. § 103(a) over Hundt in view of Srivestava.

A. Claim Limitations are not Met by the Cited References

Appellants respectfully submit that the rejection of the Claims is improper as the rejection of Claims 1-24 does not satisfy the requirements of a *prima facie* case of obviousness as claim limitations are not met by the cited reference.

The Appellants have reviewed the cited references and respectfully submit that the present invention as recited in Claims 1-24 is not rendered obvious by Hundt or Srivastava, alone or in combination.

To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). (MPEP 2143.03).

Appellants respectfully agree with the statement on page 2 of the instant Office Action, to the effect that Hundt does not teach identifying an inlined function in source code wherein the source code is for a binary executable. The instant Office Action cites Srivastava as teaching this feature at column 13, lines 42-53. Appellants respectfully disagree, for the reasons below.

At line 43 of column 13, Srivastava specifically states "[t]hese routines may be used to identify procedures of the program which may be placed in-line to improve execution performance" (emphasis added). The "routines" referred to by Srivastava are "exemplary user instrumentation and analysis routines that can be used to monitor different operating characteristics of the program" (see Srivastava column 12, starting at line 53; emphasis added). Quite clearly, these routines are being discussed in conjunction with Srivastava's Figures 4 and 5, which pertain to Srivastava's "program 100" (also referred to as procedures

100). Also, please see, for example, column 6, lines 45-51, of Srivastava, which indicates that program 100 is being instrumented for the purpose of performance monitoring.

In other words, "the program" cited at column 13, lines 42-53, of Srivastava is, unequivocally, program 100.

Importantly, Srivastava's program 100 is not "source code." According to Srivastava, source code modules 21-23 are included in program 20 not program 100. Additionally, please see Figure 2 of Srivastava, which indicates that program 100 is derived from the executable code 60, not the source code modules 21-23.

Therefore, Appellants respectfully submit that Srivastava does not teach that which it is relied upon as teaching. Specifically, Appellants respectfully submit that Srivastava does not teach or render obvious identifying an inlined function in source code wherein the source code is for a binary executable (emphasis added).

As such, Appellants respectfully submit that neither Hundt nor Srivastava, nor the combination thereof, show or suggest "identifying an inlined function in source code" as recited in independent Claims 1, 7 and 13. Accordingly, Appellants respectfully submit that the basis for rejecting Claims 1, 7 and 13 under 35 U.S.C. § 103(a) is traversed.

Furthermore, Appellants respectfully point out that in order to establish a *prima facie* case of obviousness, the prior art must suggest the desirability of the claimed invention (MPEP 2142). "[i]f the proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed amendment" (emphasis added) (MPEP 2143.01; *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984)).

Appellants respectfully submit that the version of "HP Caliper" described in the Hundt reference is a performance analysis tool for binary executables (see at least the second page of that reference, starting at about line 12). As described in the background section of the instant application (page 2, lines 5-8), a shortcoming of conventional performance analysis tools such as that described by the Hundt reference is that "even if the programmer specifies in the source code that a certain function be inlined, that does not necessarily mean that the particular function will ultimately be inlined in the binary executable by the compiler" (emphasis added). As a performance analysis tool apparently limited to binary executables, and without evidence to the contrary, the particular version of "HP Caliper" described in the Hundt reference would share the above shortcoming, and it is an object of the present claimed invention to address that shortcoming.

As previously stated, Appellants understand Srivastava to teach "[t]hese routines may be used to identify procedures of the program which may be placed in-line to improve execution performance" (emphasis added). The "routines" referred to by Srivastava are "exemplary user instrumentation and analysis routines that can be used to monitor different operating characteristics of the program" (see Srivastava column 12, starting at line 53; emphasis added). Quite clearly, these routines are being discussed in conjunction with Srivastava's Figures 4 and 5, which pertain to Srivastava's "program 100" (also referred to as procedures 100). Also, please see, for example, column 6, lines 45-51, of Srivastava, which indicates that program 100 is being instrumented for the purpose of performance monitoring.

In other words, "the program" cited at column 13, lines 42-53, of Srivastava is, unequivocally, program 100. Importantly, Srivastava's program 100 is not "**source code**" (emphasis added). According to Srivastava, source code modules 21-23 are included in program 20 not program 100. Additionally, please see Figure 2 of Srivastava, which indicates that program 100 is derived from the executable code 60, not the source code modules 21-23.

As such, Appellants respectfully submit that the combination of Hundt in view of Srivastava to obtain the features of Claims 1, 7 and 13 is improper as it would fundamentally change the method of operation of Hundt. That is, Hundt teaches (at line 12 of page 2) a performance analysis tool for binary executables and requires no special preparation or recompilation of these binaries. As such, the suggested modification of Hundt to "identifying an inlined function in source code" is improper as it would fundamentally change the method of operation of Hundt (emphasis added). Accordingly, Appellants respectfully submit that the basis for rejecting Claims 1, 7 and 13 under 35 U.S.C. § 103(a) is traversed.

In view of the combination of Hundt in view of Srivastava not showing or suggesting all of the limitations of independent Claims 1, 7 and 13, not satisfying the requirements of a *prima facie* case of obviousness, Appellants respectfully submit that independent Claims 1, 7 and 13 overcome the rejection under 35 U.S.C. § 103(a), and that these claims are thus in a condition for allowance. Appellants respectfully submit the combination of Hundt in view of Srivastava also does not teach or suggest the additional claimed features of the present invention as recited in Claims 2-6 and 19-20 that depend from independent Claim 1, Claims 8-12 and 21-22 that depend from independent Claim 7, and Claims 14-18 and 23-24 that depend from independent Claim 13. Therefore, Appellants respectfully submit that Claims 2-6, 8-12 and 14-24 also overcome the rejection under 35 U.S.C. § 103(a), and are in a condition for allowance as being dependent on an allowable base claim.

With respect to page 6 and 7 of the present Final Office Action and Specifically the Response to Arguments Section page 7 lines 4-8, Appellants respectfully note that the Office Action states, "[A]s is known, an inline function is one in source code where a compiler copies the code from the inlined function definition directly into the calling function code instead of creating a separate set of code in memory to avoid the performance overhead of a function call."

Appellants respectfully submit that the present Office Action has provided inadequate support of a finding. Moreover, Appellants respectfully submit that the basis for the "as is known" as relied on by in the present Office Action is not set forth explicitly, as required. Appellants respectfully submit that the Examiner has not stated why the <u>inline function</u> <u>**is**</u> one in <u>source code</u> (emphasis added). As such, Appellants respectfully assert that the Office Action has taken "as is known" without providing a clear and unmistakable technical line of reasoning, as required.

"It is never appropriate to rely solely on common knowledge in the art without evidentiary support in the record as the principal evidence upon which a rejection is based" (emphasis added; MPEP 2144.03(E); See *In re Zurko*, 258 F.3d 1379, 1386, 59 USPQ2d 1693, 1697 (Fed. Cir. 2001); *In re Ahlert*, 424 F.2d 1088, 1092, 165 USPQ 418, 421 (CCPA 1970)). The "assessment of basic knowledge and common sense that is not based on any evidence in the record lacks substantial evidence support" (MPEP 2144.03(A); *In re Zurko*, 258 F.3d 1379, 1385, 59 USPQ2d 1693, 1697 (Fed. Cir. 2001)). In particular, "[i]f such notice is taken, the basis for such reasoning must be set forth explicitly. The examiner must provide specific factual findings predicated on sound technical and scientific reasoning to support his or her conclusion of common knowledge" (MPEP 2144.03(B); see *In re Soli*, 317 F.2d 941, 946, 37 USPQ 797, 801 (CCPA 1963); see also *In re Chevenard*, 139 F.2d 711, 713, 60 USPQ 239, 241 (CCPA 1943)).

Accordingly, Appellants respectfully that the Examiner provide adequate evidence in the form of an affidavit in support of the finding of "as is known", in accordance with 37 CFR § 1.104(d)(2).

Regarding Claims 20, 22 and 24

Appellants respectfully contend that Hundt does not show or suggest "reading source correlation information from within said binary executable; and obtaining start and end addresses for said inlined function using said source correlation information" as recited in Claims 20 and 22 and as similarly recited in Claim 24. The final rejection cites Section 4.1 of the Hundt reference, but Appellants find no teaching in either Section 4.1 or in Hundt as a whole with regard to the limitations of Claims 20, 22 and 24. Hundt mentions that function entry points are identified by analysis of unwind information tables (e.g., exception tables), procedure lookup tables, and the symbol table (please see step 2 in Section 4.1 of Hundt). Even presuming that the functions referred to by Hundt are inlined functions and that the entry points referred to by Hundt are addresses for the functions, Appellants respectfully assert that Hundt does not show or suggest using source correlation information to obtain addresses for inlined functions. Thus, Appellants respectfully submit that an essential element needed for a *prima facie* rejection of Claims 20, 22 and 24 is missing.

In summary, Appellants respectfully submit that the Examiner's rejections of the Claims are improper as the rejection of Claims 1-24 does not satisfy the requirements of a *prima facie* case of obviousness as claim limitations are not met by the cited reference. Accordingly, Appellants respectfully submit that the rejection of Claims 1-24 under 35 U.S.C. §103(a) is improper and should be reversed.

Conclusion

Appellants believe that pending Claims 1-24 are directed toward patentable subject matter. In particular, Appellants believe that pending Claims 1-24 are patentable over Hundt in view of Srivastava. As such, Appellants submit that Claims 1-24 are patentable over the cited references.

Appellant respectfully requests that the rejection of Claims 1-24 be reversed. The Appellant wishes to encourage the Examiner or a member of the Board of Patent Appeals to telephone the Appellant's undersigned representative if it is felt that a telephone conference could expedite prosecution.

Respectfully submitted,
Wagner Blecher LLP

Dated: 9/26/07

John P. Wagner, Jr.
Registration No.: 35,398

Wagner Blecher LLP
Westridge Business Park
123 Westridge Drive
Watsonville, CA 95076

Phone: (408) 377-0500
Facsimile: (408) 722-2350

VIII. Appendix - Clean Copy of Claims on Appeal

1. (previously presented) A computer-implemented method for examining an inlined function using a performance analysis tool, said method comprising:

identifying an inlined function in source code, wherein said source code is for a binary executable;

inserting a breakpoint at the start of said inlined function in said binary executable; and

replacing said inlined function with a long branch to a shared memory probe code sequence.

2. (Original) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 further comprising:

creating a data structure which maintains location information for said inlined function and information related to said desired task for said inlined function.

3. (previously presented) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 further comprising:

using said performance analysis tool to perform instrumentation on said inlined function.

4. (previously presented) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 further comprising:

using said performance analysis tool to perform mapping of samples to said inlined function.

5. (Previously Presented) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 wherein said performance analysis tool is comprised of an instrumentation application.

6. (Previously Presented) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 wherein said performance analysis tool is comprised of a sampling application.

7. (previously presented) A computer-readable medium embodying instructions that cause a computer to perform a method for examining an inlined function using a performance analysis tool, said method comprising:

identifying an inlined function in source code, wherein said source code is for a binary executable;

inserting a breakpoint at the start of said inlined function in said binary executable; and

replacing said inlined function with a long branch to a shared memory probe code sequence.

8. (Previously Presented) The computer-readable medium of Claim 7 further comprising instructions that cause said computer to perform said method further comprising:

creating a data structure which maintains location information for said inlined function and information related to said desired task for said inlined function.

9. (previously presented) The computer-readable medium of Claim 7 further comprising:

using said performance analysis tool to perform instrumentation on said inlined function.

10. (previously presented) The computer-readable medium of Claim 7 further comprising:

using said performance analysis tool to perform mapping of samples to said inlined function.

11. (Previously Presented) The computer-readable medium of Claim 7 wherein said performance analysis tool is comprised of an instrumentation application.

12. (Previously Presented) The computer-readable medium of Claim 7 wherein said performance analysis tool is comprised of a sampling application.

13. (previously presented) An apparatus for examining an inlined function using a performance analysis tool, said apparatus comprising:

means for identifying an inlined function in source code, wherein said source code is for a binary executable;

means for inserting a breakpoint at the start of said inlined function in said binary executable; and

means for replacing said inlined function with a long branch to a shared memory probe code sequence.

14. (Original) The apparatus of Claim 13 further comprising:

means for creating a data structure which maintains location information for said inlined function and information related to said desired task for said inlined function.

15. (previously presented) The apparatus of Claim 13 further comprising means for performing instrumentation on said inlined function.

16. (previously presented) The apparatus of Claim 13 further comprising means to perform mapping of samples to said inlined function.

17. (Previously Presented) The apparatus of Claim 13 wherein said performance analysis tool is comprised of an instrumentation application.

18. (Previously Presented) The apparatus of Claim 13 wherein said performance analysis tool is comprised of a sampling application.

19. (Previously Presented) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 wherein said shared memory probe code sequence saves registers, executes the original bundle of said inlined function, restores said registers, and jumps back to said computer code.

20. (Previously Presented) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 wherein said inserting further comprises:

reading source correlation information from within said binary executable; and

obtaining start and end addresses for said inlined function using said source correlation information.

21. (Previously Presented) The computer-readable medium of Claim 7 wherein said shared memory probe code sequence saves registers, executes the original bundle of said inlined function, restores said registers, and jumps back to said computer code.

22. (Previously Presented) The computer-readable medium of Claim 7 wherein said inserting further comprises:

reading source correlation information from within said binary executable code; and

obtaining start and end addresses for said inlined function using said source correlation information.

23. (Previously Presented) The apparatus of Claim 13 wherein said shared memory probe code sequence saves registers, executes the original bundle of said inlined function, restores said registers, and jumps back to said computer code.

24. (Previously Presented) The apparatus of Claim 13 further comprising:

means for reading source correlation information from within said binary executable code; and

means for obtaining start and end addresses for said inlined function using said source correlation information.

IX. Evidence Appendix

No evidence is herein appended.

X. Related Proceedings Appendix

No related proceedings.